

# MELV1N

TECH INTERVIEW SUPERPOWER DOCUMENT

Thanks for downloading this 'freebee' document from [my article](#) to help you hire developers.

## Goal

This document is to help you ask questions on a certain technical level when you don't have the technical knowledge to do so. It gives you technical superpower when you need it the most.

## Structure

- This document is chopped up in sections for a specific (technical) area. You can easily select which area's you want to address and find appropriate interview questions. Just pick the questions that are most suitable for the person sitting in front of you.
- Each question has its answer. It can seem overwhelming with technical jargon. Don't worry, its just to be accurate and getting used to those words.
- Each answer has an extra non-technical explanation. It will help you understand what words in the answer to look for during a conversation.
- Most of the structure is based on the 'Skills pyramid of a developer'. If you don't know what that is, please read my article first for more explanation (link above).
- Don't worry if you hear a lot of 'Star wars language' in the candidate's answer. Just focus on the parts in the answer you can connect from the explanations.

This document is licensed under [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#).

# Vision

---

## 1. Why do you want to work here?

Answer: anything aligned with the company mission.

*Explanation: It's the most important question of all. If you can't answer clearly why you want to work here, don't bother hiring.*

## 2. What value do you think you'll deliver to the team and company?

Answer: Core product value and improve knowledge for yourself and others.

*Explanation: Building a great product goes hand in hand with improving along the way. You do that by contributing your knowledge to others and vice-versa.*

## 3. What do you want to learn?

Answer: Clear view of personal learning goals.

*Explanation: A good developer has clear learning goals on a technical and career level. He or she is picking a company partly because they can improve knowledge and skills to go to their next-level.*

## 4. Describe yourself within the company after 1 year. What do you want to have achieved?

Answer: Clear view of job expectations.

*Explanation: Most initial contracts are for about a year. You want to have a clear understanding about what someone is expecting from their adventure at your company and if this matches with reality. Don't hire someone who clearly has unrealistic expectations that your company can't provide.*

## 5. Who do you think is paying for your salary here?

Answer: Customers.

*Explanation: Understanding customers pay the bills and you work for them, not the CEO.*

# Social & Communication

---

## 1. Describe the last time you had a conflict and how you resolved it.

Answer: Resolved it in a manner that exposes being able to reflect on yourself and actively take on the role of resolving conflict without any relationship damage. Also talking to someone immediately after have a conflict moment instead of waiting days or more ignoring someone (neglecting).

*Explanation: Developers can have a lot of conflict (remember the stubborn part?) from technical decision making. It's extremely important to resolve conflicts fast to prevent any snowball effect between developers.*

## 2. How would friends describe you?

Answer: Some critique, characteristics he/she is not ashamed of but may not be the most positive of all.

*Explanation: This may seem a bit odd but there is an important element. If you have deep connection with friends, they'll also give you reflections of yourself where you may need to improve and talk about it. You're looking for mentions of characteristics that are not super positive but portray honest feedback that someone takes as a challenge to improve. Being able to have such connection with people exposes real social efforts. You can't lead people without being able to process honest head-to-head feedback.*

## 3. What do you do for fun?

Answer: Any real hobbies away from the computer.

*Explanation: The specific activities are not relevant. Development is a high-performance job with constant pressure. Your brain needs downtime to relax and re-energize itself to be able to cope with job performance level. It needs to be clear someone is spending downtime and is not enslaved to his coding editor all day everyday.*

# Tech

---

## **1. Programming: Can you describe the difference between a class and a method?**

Answer: A class is basically a definition (or template) of a method and variables. Those methods/variables create something called objects.

*Explanation: Fundamental programming question. It's easy to explain: a class is like a blueprint of a house. You can use the blueprint to take the correct building blocks (methods and variables) to build a house (object).*

## **2. Programming: Can a function exist without a class?**

Answer: Yes

*Explanation: It's a trick question. Juniors can be tricked with this one. Functions actually exist longer than classes in software engineering history. They have no direct relationship with each other.*

## **3. Programming: Explain why in most cases a for loop is preferred over a while loop?**

Answer: Most loop cases don't require a condition to stop the loop. You want to prevent errors that would cause infinite looping (= breaking code) due to invalid conditioning. Most sets of data you want to loop over are set so a for loop is preferred.

*Explanation: Also basic question for most programming languages. Everything you see in a list or set of data is produced by code that needs to be repeated to execute over every value of that list (like a list of numbers). That process of repeating code over a list is called a loop. You need to look for the condition part in his answer.*

#### **4. Databases: Explain the biggest difference between MongoDB or Redis**

Answer: MongoDB is a schemaless database where Redis is a key-value store.

*Explanation: More overall questions about databases. If you need to pick new technology, you need to know what some solutions are good at and what they do. MongoDB and Redis are well-known, there is no excuse not knowing this simple difference. Just look for the words 'MongoDB schemaless' and 'key-value store'.*

#### **5. Databases: What does a temporary table in MySQL do and why are they considered bad practice for normal application operation?**

Answer: Temporary tables are used in multiple use cases (which are ok) like handling batches of data to process temporarily in a certain scope or to cope with complex querying. Temp tables impact performance heavily when you table size is expanding heavily while the query remains complex or gets even more complex.

*Explanation: Very detailed databases question. Just look for the performance part and query complexity aspects.*

#### **6. Databases: In a standard master-slave database server setup, which one would do the read and which the write queries?**

Answer: The slave database servers do all the reads, the master does all the writes.

*Explanation: Very standard setup for a database layer. Slaves are basically multiple database machines to cope with the volume of requests. Read means it only needs to get data for a request (like getting an username). Write means it needs to change data for a request (like changing an username).*

#### **7. Testing: What is the difference between end-to-end testing and unit testing?**

Answer: Unit testing is testing a piece of code (or code module) in the backend to check if it executes correctly. End-to-end testing is tested by an automated browser system to check from the frontend (the actual page) if all the user related actions work.

*Explanation: This will reveal if the candidate has real testing experience. Look for the words 'unit testing code backend' and 'frontend end-to-end user testing'.*